StarMail

StarMail 4.1.0

- Technical Manual

Efficient e-mail communication with your Target Group

StarMail 4.1.0 Technical Manual v1.0. Copyright 2007 Netstar AB.

Netstar AB Box 3415 103 68 Stockholm, Sweden
Phone. +46 (0) 8 5000 98 70 Fax. +46 (0) 8 5000 98 71 products@netstar.se www.netstar.se

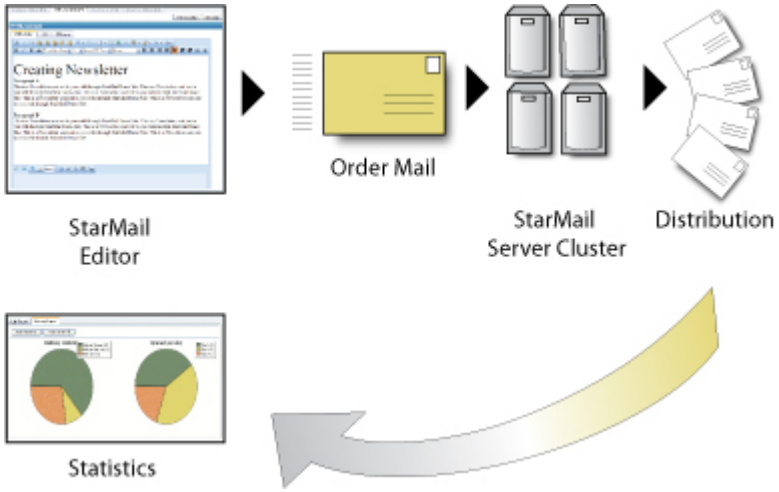
NETSTAR

Table of Content

1. About StarMail 3
2. Managing and creating recipients lists 3
   - 2.1.1. Site Partitioning 4
   - 2.2. Source – Recipient List 4
     - 2.2.1. Recipient class 4
     - 2.2.2. RecipientContainer class 5
     - 2.2.3. Creating a Source 5
     - 2.2.4. Sample Code 5
   - 2.3. Standard - Recipient List 6
     - 2.3.1. Import 6
     - 2.3.2. CSV files 6
     - 2.3.3. XML 7
3. Order Mail 7
   - 3.1. Specification of StarMail Order Mail 2.0 8
     - 3.1.1. Options 8
     - 3.1.2. Mail data HTML 9
     - 3.1.3. Mail Data Text 9
     - 3.1.4. Email addresses and variables 9
     - 3.1.5. Encryption 10
     - 3.1.6. Headers 11
4. Statistics 12
   - 4.1. Web service 12
     - 4.1.1. SOAP details 12
     - 4.1.2. Statistic setup 13
     - 4.1.3. What kind of Statistics is it possible to get. 14
     - 4.1.4. Bounce Management 14
   - 4.2. StarMail Configuration File 14
5. System Settings 16
   - 5.1. Users and User Groups 16
   - 5.2. Administrative rights 18
   - 5.3. Site Partitioning 21
   - 5.4. Globalization 22
   - 5.5. Licensing 23
6. Requirements 24
   - 6.1. Knowledge 24
   - 6.2. System requirements 24
7. Related products 24
8. Help us get better 24
9. Appendix 1. StarCommunity Source Sample Code 25

1. About StarMail

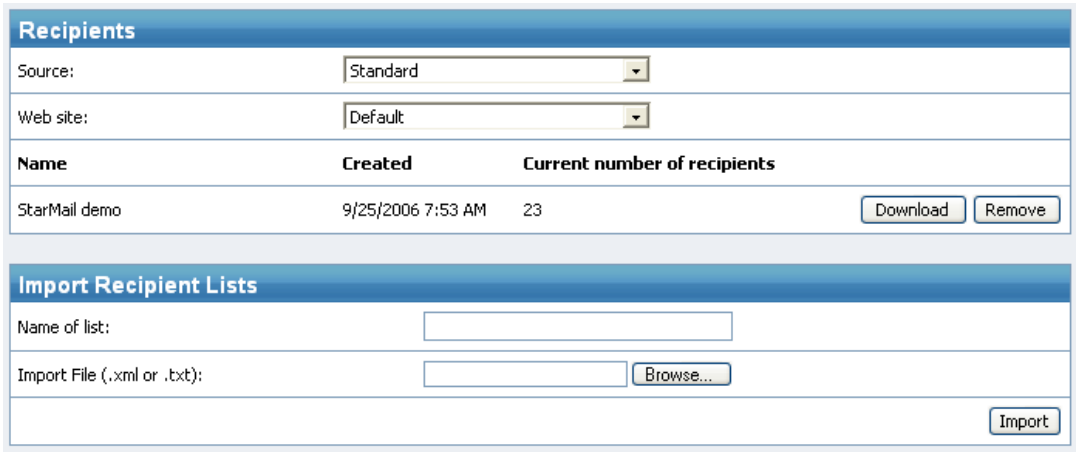
StarMail is powerful software for distribution of emails and newsletters. StarMail is designed to handle very large numbers of email recipients. StarMail consists of a client (administration) part and a server part, both of which are described in this Technical Manual. In the client part, the StarMail Editor, the email or newsletter is created and the distribution is initiated. An order email with content and recipient data is sent to our dedicated StarMail Server Cluster. This cluster of servers then distributes the emails or newsletters without consuming valuable power from the sender's IT environment. Detailed real-time statistics for each newsletter is available instantly.



- The web based StarMail Editor is where the message is created. HTML mails can be composed by using the built in html editor or by selecting a page already created in EPiServer, DocStar, or by providing a URL.
- An order email is sent to the StarMail Broker, which calculates the most efficient way to distribute the message to the recipients via the servers in its cluster, considering current load and queue length for each distribution server.

2. Managing and creating recipients lists

From a developers perspective all recipient sources inherits and implements the IRecipientSource Interface. Each recipient source can provide any number of recipient lists (of the type RecipientContainer). One implementation that supports recipient list import is included with StarMail Editor and does not need to be implemented by the developer. This recipient source is called "Standard" and is used whenever a list is to be imported.



However, in many cases there is need for dynamic recipient lists, to integrate StarMail with an existing system where the recipient information resides.

Most commonly, such a list is generated by database queries. To do this you need to implement the IRecipientSource Interface. Description of this and samples can be found in section 2.2.

2.1.1. Site Partitioning

For larger organizations, it is possible to create different "sites" that selected users or groups can be given access to. See the section System Settings – User and user groups.

2.2. Source – Recipient List

To implement your own recipient source you need to implement the IRecipientSource interface.

2.2.1. Recipient class

The Recipient class is defined in the StarMail.Core namespace. The Recipient class defines the following public properties:

- Guid **ID** - The unique identifier of the recipient
- string **Email** - The email address of the recipient
- AttributeCollection **Attributes** - The collection of attributes of the recipient. These attributes are used for substitution of variables in the mail body. The variables in the mail body must be in the format §a§, §b§, §c§ etc. and are substituted by the corresponding attribute where the order of attributes in the attribute collection coincides with the alphabetic order of the variable name.
- bool **IsUnsubscribed** - Gets whether the user has been marked as unsubscribed
- Guid **ContainerId** - Gets the unique identifier of the RecipientContainer that this Recipient is connected to.

2.2.2. RecipientContainer class

The RecipientContainer class represents a specific list of recipients. One of the properties of a RecipientContainer is an implementation of the IRecipientSource interface that the RecipientContainer was provided by.

2.2.3. Creating a Source

This section describes the properties and methods defined in the IRecipientSource Interface in the StarMail.Core namespace that needs to be implemented.

- Guid **ID** – The unique identifier for this source. This must be constant for any given source.
- bool **SupportsImport** – In most cases false since an implementation that supports this is already shipped with the StarMail Editor ("Standard" source).
- string **Name** – The name of this source. This must be constant for any given source.
- RecipientContainer **ImportRecipients** – If SupportsImport is set to false, the implementation should just return null.
- RecipientContainerCollection **GetRecipientContainers** – should return a recipient container collection. (If the provided Site is not applicable, return your list of RecipientContainers when the Site is null.)
- Recipient **GetRecipient** – Should return the recipient for the supplied Guid
- RecipientCollection **GetRecipients** - Should return the recipients for the supplied RecipientContainer
- void **UnsubcribeRecipient** – Unsubscribes or re-subscribes the supplied recipient.
- RecipientContainer **GetRecipientContainer** – Should return the recipient container with the supplied Guid or name.
- void **RemoveRecipientContainer** – Should remove a recipient container with the supplied guid. If removal of RecipientContainers is not supported, just return.

The compiled assembly should be put in the bin directory on the web server.

2.2.4. Sample Code

When running StarMail together with StarCommunity or EPiServer 4.x there are options to install implementations of IRecipientSource for these during the StarMail Installation. The StarCommunity source selects all current StarCommunity users with the subscription user attribute set. The name of the attribute can be specified in the configuration file. The EPiServer source selects all current EPiServer users. The sample code in appendix 1 is an excerpt from the StarCommunity source. When creating your own sources, you may want to start by modifying the GetRecipients method. If you are running StarCommunity you can use the user search handler to get the recipients (as in the sample). However, you can of course create customized database queries to get recipients from any database or other data source accessible.

2.3. Standard - Recipient List

2.3.1. Import

StarMail can import E-mail address lists from XML and CSV text files (semi colon separated).

2.3.2. CSV files

Most common is importing Email lists from Excel.

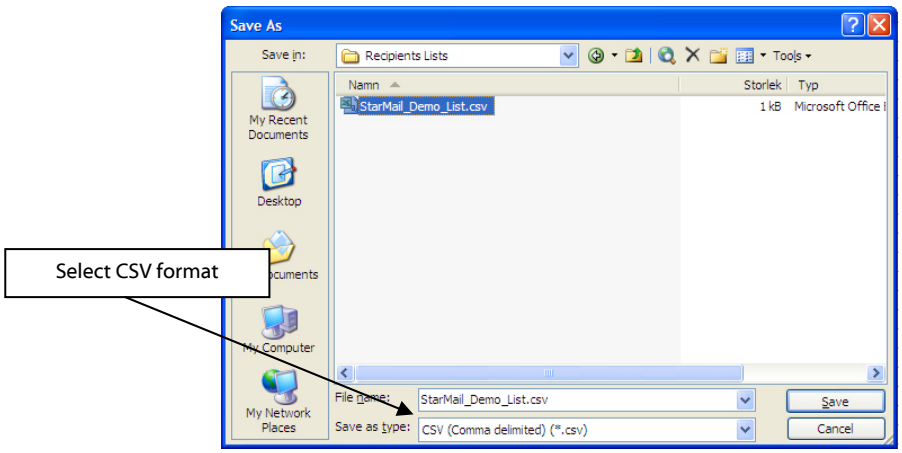
When importing email addresses from Microsoft Excel the file needs to be saved in a specific format. Please see instructions below.

	A	B	C	D
1	**E-mail**	**Name**	**Sur Name**	**Title**
2	emailA@netstar.se	Andreas	Stjernström	Partner Manager
3	emailB@netstar.se	Gustav	Lagercrantz	CEO
4	emailC@netstar.se	Elvis	Presley	The King
5	emailD@domain.com	Tiger	Woods	Golf Pro
6				
7				
8				

The first row is a header row and is not imported to StarMail. The first column is always used as the email address. It is possible to import a list that consists of only one column with email addresses. In the example above, there are additional columns with variables such as Name, Sur Name and Title. These variables can be used for different needs and are used to personalize each email sent from StarMail.

StarMail handles up to 20 variables for each recipient list.

Save the Excel sheet as a CSV-file (Semi colon delimited) and make sure that the file has the format UTF-8. If it is not possible to make sure that the CSV-file is in UTF-8 format, open the file with a regular text editor, such as Notepad, and save the file again making sure that the file is in UTF-8 format.



2.3.3. XML

An email list can also be imported from an XML-file. See below for an example of a compatible XML-file:

```
<?xml version="1.0" encoding="utf-8" ?>
<starMailImport>
     <attributes>
            <attribute id="1">attributeName1</attribute>
            <attribute id="2">attributeName2</attribute>
            <attribute id="3">attributeName3</attribute>
     </attributes>
     <recipient>
            <email>email1@mail.com</email>
            <attributeList>
                  <attribute id="1">attribute value for attr 1</attribute>
                  <attribute id="2">attribute value for attr 2</attribute>
                  <attribute id="3">attribute value for attr 3</attribute>
            </attributeList>
     </recipient>
     <recipient>
            <email>email2@mail.com</email>
            <attributeList>
                  <attribute id="1">attribute value for attr 1</attribute>
                  <attribute id="2">attribute value for attr 2</attribute>
                  <attribute id="3">attribute value for attr 3</attribute>
            </attributeList>
     </recipient>
     <recipient>
            <email>email3@mail.com</email>
            <attributeList>
                  <attribute id="1">attribute value for attr 1</attribute>
                  <attribute id="2">attribute value for attr 2</attribute>
                  <attribute id="3">attribute value for attr 3</attribute>
            </attributeList>
     </recipient>
</starMailImport>
```

3. Order Mail

The distribution is initiated when the StarMail Server cluster receives an order mail. The order mail is created either according to below specification or with the StarMail Editor (the client part of StarMail).

The order mail includes the actual email content, subject, sender information, recipients and, etc.

The order mail is sent to ***"account"*@starmail.netstar.se**

3.1. Specification of StarMail Order Mail 2.0

Below is a an example and specification of an order mail that complies to StarMail Order Mail Version 2.0:

```
----options----
subject: Subject Text
sender: name@domain.com
sendername: The Name
guid: f1e5693f-49b0-40f7-b526-8d3befc02ebc
senddate: 2006-12-30 24:30
----endofoptions----
----maildata----
<html>
Hi §a§ §b§!<br>
<br>
Here is the content of the  <i>email or Newsletter</i>.
<br>
Your title is §c§
</html>
----endofmaildata----
----maildatatext----
Hi §a§!
Here is the content of the email or Newsletter.
----endofmaildatatext----
----htmladdr----
emailA@netstar.se;Andreas;Stjernström;Partner Manager
emailB@netstar.se;Gustav;Lagercrantz;CEO
emailC@netstar.se;Elvis;Presley;The King
emailD@netstar.se;Tiger;Woods;Golf Pro
----endofhtmladdr----
----textaddr----
emailE@netstar.se;Andreas;Stjernström;Partner Manager
emailF@netstar.se;Gustav;Lagercrantz;CEO
emailG@netstar.se;Elvis;Presley;The King
emailH@netstar.se;Tiger;Woods;Golf Pro
----endoftextaddr----
```

3.1.1. Options

In the part of the order mail named ----Options---- following parameters could be set.

```
----options----
subject: Subject Text
sender: Name@domain.com
sendername: Name
guid: f1e5693f-49b0-40f7-b526-8d3befc02ebc
senddate: 2006-12-30 24:30
charset: iso-8859-1
----endofoptions----
```

"**Subject**" sets the Subject text for the message

"**Sender**" Sets the sender E-mail adress.

"**Sendername**" Sets the name of the sender.

"**GUID**" sets the Globally Unique Identifier of the specific newsletter to prevent the same mail from being sent out more then once. This replaces the deprecated option "Version".

"**Senddate**" sets when the distribution starts. With this setting it is possible to schedule a distribution. Format for senddate is YYYY-MM-DD HH:MM. Setting this is optional (the entire row should be excluded if not used).

3.1.2. Mail data HTML

In the part of the order mail named ----maildata---- the content of the mail is included. The maildata is only delivered to the email addresses in the order mail ----htmladdr----

```
----maildata----
<html>
Hi §a§ §b§!<br>
<br>
Here is the content of the  <i>email or Newsletter</i>.
<br>
Your title is §c§
</html>
 ----endofmaildata----
```

In the example above, variables are being used. The variables are inserted in the email by writing § before the variable name for example §a§. The variables are assigned names automatically; a, b, c, etc. Below are two examples of how the resulting email from the example would look like to the end user:

Email 1
Hi Andreas Stjernström
Here is the content of the *email or Newsletter*.
Your title is Partner Manager

Email 2
Hi Gustav Lagercrantz
Here is the content of the *email or Newsletter*.
Your title is CEO

3.1.3. Mail Data Text

The part of the order mail named ----maildata---- works the same way as Maildata exept that the content could not be html. The maildata is only delivered to the email addresses in the order mail ----textaddr----

```
----maildatatext----
Hi §a§!
Here is the content of the email or Newsletter.
----endofmaildatatext----
```

3.1.4. Email addresses and variables

In the part of the order mail named ----htmladdr---- and ----textaddr---- the addresses and variables for the distribution is included. One row for each recipient, containing email addresses and attribute values, separated by semicolon.

```
----htmladdr----
emailA@netstar.se;Andreas;Stjernström;Partner Manager
```

```
emailB@netstar.se;Gustav;Lagercrantz;CEO
emailC@netstar.se;Elvis;Presley;The King
emailD@netstar.se;Tiger;Woods;Golf Pro
 ----endofhtmladdr----
----textaddr----
emailE@netstar.se;Andreas;Stjernström;Partner Manager
emailF@netstar.se;Gustav;Lagercrantz;CEO
emailG@netstar.se;Elvis;Presley;The King
emailH@netstar.se;Tiger;Woods;Golf Pro
----endoftextaddr----
```

Below illustrates how the variables for the recipient on the first row are interpreted:

Email address = emailA@netstar.se
Variable A - §a§ = Andreas
Variable B - §b§ = Stjernström
Variable C - §c§ = Partner Manager

3.1.5. Encryption

StarMail Order Mail Specification 2.0 is introducing support for RSA/AES-based encryption. If the <BrokerPublicKey> parameter in StarMail.config contains a valid public key at StarMail Editor initialization; all outgoing order mails are encrypted.

Here is an example of an encrypted ordermail:

```
-----BEGIN ORDERMAIL MESSAGE-----
Version: StarMail Client 4.1.0.0

cdRBVQS7uTf0MdUi0Q+PF6BOR9Lm3Ai01nYuvRhD1gh0hDjCIvDl1rFx5Ixkwps/bonMFg3neHIO
LvTtokpdxefTO/avb3RsbpIZsa7ngD2xHUPyvu6AOTefniu+G+Ihob99O3JwwjCHf+AC76Ka7Jf1
9XcMafTYeiHUCN27NU646cCGkN3bbZGzdAWst0t+/XXvu6jAMiAcAsbJR1RN2X6wBe2fT06hjg0T
89ewNKg3Ks2+d/gtvM55EsbnHi0NksVV6NbR5OP4OYSP2PYUJ3NLX6xWzHKBPhBaFUYFxhtqcyGe
ru1X3XKQP/DNpqYdnMd1wovZxcW/kS1EJWQQqvTmo2dQQV1FYLwlDNkLsuuVwizRBrMJ9k+ZO5tP
B95FnlNESpgz/EP5OYd+49Di+xaTArE3DQmn87x1XYgXMRd6muhKn36D1kcGziPep59KzX1Jt0uc
HMzkZB0s/jxifU9pi2QWa7w16AgcNxebhBPtPUWj0J5l4ChwuZiFFoCGFfXci1dM21j/JOGvlVb6
Sq7E/9aHTXwi5LUS3MocGhTCak65O3jFzmXQsFq+C2sjIGAzdkd78MTFdzTSgoT/sP3XWeUqVSiQ
gVrt9aZQJXVK3QqX9kdQwHvZeKnrtfXqPDi7sexw4Aze3ElrZA5eZcPNafSzVzQBXngKI3d9SU4/
0bsOgnvN3OmHEKkvmrtSARwLbJb8aPkvpjBHXbJ/HWlLK2KExk+cHFmuOvg6Z+Rjqb65AZG3CN3v
eyyieqjyATSLdJ0Kusn6EPHGowcbnG/T75y45jcIOhX8qII0hWT1qtPfMjyhFlgPRJvJyiT8oNLr
bmffCSWwYWuUI5TSgaCmrZJ+Oy5SiKaoaQN/Px5lq8M93kMO8DWM5tMPNbmMRa6XaRZTshhEex/U
Kisp0OdT1A6THEjcrs5m+0Yufy3rgw0gOp6QL4aZGyRlifhEKlc2+o9FKnD/Dnma69C9m61WYm2j
AqGPzf9at/Vv5eEdZcgUkWrMKGCvSrXMbCS7DOPFQwPhPVstMe6xkFA2amY+LQtm4IXF1EwOF8xx
ndq/pWRXLzFqfYl9sxquVZX2eEJgd7KdINghyg9RnL1COQ==
-----END ORDERMAIL MESSAGE-----
```

```
-----BEGIN ORDERMAIL KEY-----
giLwNtgaVxYyspoYsjLt74/XfALLwm9JSXN8Z6Ynzf+2LC+Xcirln/oktq4fjDBgFwdgpqu6im0x
vM/GyrDaE9Ebzf946oWvFdM4nmWxKpZX/ZPLQcDXhO+yg6n0upzkA0nhQR5hUEaPtg4Bwt6uUf7c
kltM+dxOKjk8k0L1p1lWaRjWZdWP/5jdtUrg0aYIVLT6pbbWPhhP/Y9wrdKIEkrhPci+0ow0wi7E
e+vN62dcFTyejSi2iOYwk9ZbEmiVpqukKyDYPJMNylh+GrNOU/rIbdJMbYkXSY2Q4esowyuHDOkw
/cOKxfrjvUVK116HLLl8KzxhQbL5AI0EvQ9J0sC5F/Ar697BU6XmDE3ETiC+n9cg9E8saJwv9WVq
SCKBbur1oOgiLQO9G8NGVYaJ3Z8EDIg1QBExcF0tOfbeHjo1QaJ20Bg1rQJimzwvkOz+otpjFngL
uKq+J4j3wJmx3nEW5fT3XqT1RKVftsPBWwJgA1cjcIjqXuVLkqESGNKf+XjDuTw1SSyXZbPVwxzc
+dbwsemSy1mI6enFaW+6oH3lO8cuKXUihDRAH/JCelLnPBUKq8hqt7uDd6rA5oFWJuSP45O8UxLr
Z9+fm2AZny+Y6memYcORjhXvTYliWE5xaVtjvSvcx/2F2HMctg2LflXJ3tilvI/IBPmMAlrrtIo=
-----END ORDERMAIL KEY-----
-----BEGIN ORDERMAIL IV-----
cGIC2K8B3ZnMzGy5Lqt3JlYbsfMXB7N/T+L0i6Jz1t2zHkISQN/Ej6OwZ5+mmgWG/oXDdbhnhGVJ
M9PV6dzJiKBjCZqhCKiw6uqkemqwofM+8herk7CzsMaQ3u+pvO0G2jsBlTq8AeQx/OPhQQk/eVOs
6LO2T9iEIdywA4K09gWyuxwxMa7bAwOuKZMzweYvcfZ/tV14P3wNjNicfpop6ew7dss9RDEO4Vtp
GIy3Mnfdjgg5J0sIBH2iwO2ySwBpTSCSKe8Yh/Vdl0KPy2ObkAdA3I13u2kP+jWFTN8GM6lQcAyh
zh8GnQLSsIfhduuBJH1rtaOYgRza6NwQa9MDD177o0iomps0RgG9nfX56kEX0abl3I5lElnO1cek
qtWtZJpGbkdpZ19HdKH6hiVyfhAsm01H/dXlWLA0SqOBQtAC7QS9mgepSBM7TqmJ/0W2mjemuBK7
a7aOV3l071Oho6rA2XDL5oW8QwzMBWzIejP1fFQE65AWDqYOooWSN9W6GNeNIwbD8fVmp53LCGdr
EdV60B3QDQfKNBhvhOVvSPqoUqlEYKOLBbTxF9OynSQpnLmQoh3UmILRwND2o63MKiGHbznRZW7L
b/8zplY7gccjGWrO4gRPhW82Fv/QpVhOo929fUnPorexXGzc2dMrhJugjaRDNIWmIhFeBv5KUj0=
-----END ORDERMAIL IV-----
```

StarMail creates a unique symmetric key and uses it to encrypt the ordermail content (found starting at `-----BEGIN ORDERMAIL MESSAGE-----`). The symmetric key and initialization vector is then encypted with the custom public RSA key of the StarMail installation and supplies it for decryption on the server side, as shown in example above.

3.1.6. Headers

From StarMail Order Mail Specification 2.0, version number is defined in the mail header as parameter “StarMail-OrderMail-Version”.

4. Statistics

4.1. Web service

The Web Service section is only for systems that do not have the Starmail Editor installed.

If you are not using the StarMail Editor but want to handle bounces you need to implement the following to web service methods, which are called from the StarMail Server Cluster:

void AddBounces(int messageId, string[] emails)

void AddBounce(int messageId, string email)

AddBounces accepts an array of email addresses (strings) which are to be marked as bouncing. The messageId is the version provided in the order mail.

AddBounce works just like AddBounces, but accepts a single email address.

4.1.1. SOAP details

The WSDL specification file for the web service is available at the Netstar partner web site.

Here are example calls for the web service methods returning bounces for a newsletter using StarMail Order Mail Specification 2.0 or later.

Example 1a

```
POST /Netstar/StarMail/StarMail.asmx HTTP/1.1
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://netstar.se/StarMail/AddBouncesByGuid"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AddBounces xmlns="http://netstar.se/StarMail/">
      <messageId>f1e5693f-49b0-40f7-b526-8d3befc02ebc</messageId>
      <emails>
        <string>email1@domain.com</string>
        <string>email2@domain.com</string>
      </emails>
    </AddBounces>
  </soap:Body>
</soap:Envelope>
```

Example 1b

```
POST /Netstar/StarMail/StarMail.asmx HTTP/1.1
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://netstar.se/StarMail/AddBounceByGuid"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AddBounce xmlns="http://netstar.se/StarMail/">
      <messageId>f1e5693f-49b0-40f7-b526-8d3befc02ebc</messageId>
      <email>email1@domain.com</email>
```

```
    </AddBounce>
  </soap:Body>
</soap:Envelope>
```

Here are example calls for the web service methods returning bounces for a newsletter using a StarMail Order Mail Specification prior to 2.0.

Example 2a

```
POST /Netstar/StarMail/StarMail.asmx HTTP/1.1
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://netstar.se/StarMail/AddBounces"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AddBounces xmlns="http://netstar.se/StarMail/">
      <messageId>17</messageId>
      <emails>
        <string>email1@domain.com</string>
        <string>email2@domain.com</string>
      </emails>
    </AddBounces>
  </soap:Body>
</soap:Envelope>
```

Example 2b

```
POST /Netstar/StarMail/StarMail.asmx HTTP/1.1
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://netstar.se/StarMail/AddBounce"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AddBounce xmlns="http://netstar.se/StarMail/">
      <messageId>17</messageId>
      <email>email1@domain.com</email>
    </AddBounce>
  </soap:Body>
</soap:Envelope>
```

4.1.2. Statistic setup

Statistics for number of opened emails and clicks are collected through the /Netstar/StarMail/GoMail.aspx page. This page has to be publicly available for the statistics collection to work.

If the StarMail Editor is not used, it is necessary to implement such a page to make it possible to get statistics. Note that links in the HTML mail will need to be modified to go to this page prior to sending the order mail. The “gomail” page itself should to be set up to log the request and then redirect to the original destination.

4.1.3. What kind of Statistics is it possible to get.

Opened messages

Unique opened messages

Unknown

Export opened

Clicks

Export clicks

When the messages opened

4.1.4. Bounce Management

The StarMail server collects information about emails that bounce (not deliverable). The information is sent back to the sender using a SOAP Web Service interface. It is up to the sender to decide action on e-mail addresses that bounce. For example, if an address bounces more than two times it could be removed from the recipient list.

When using the StarMail Editor and its built in bounce reporting system, bounces are recorded in the table tblStarMailBounce.

4.2. StarMail Configuration File

The data in the configuration file is created when installing StarMail. To change the StarMail configuration after the installation it can be done in the StarMail Configuration file.

The configuration file is located at /Netstar/StarMail/StarMail.config

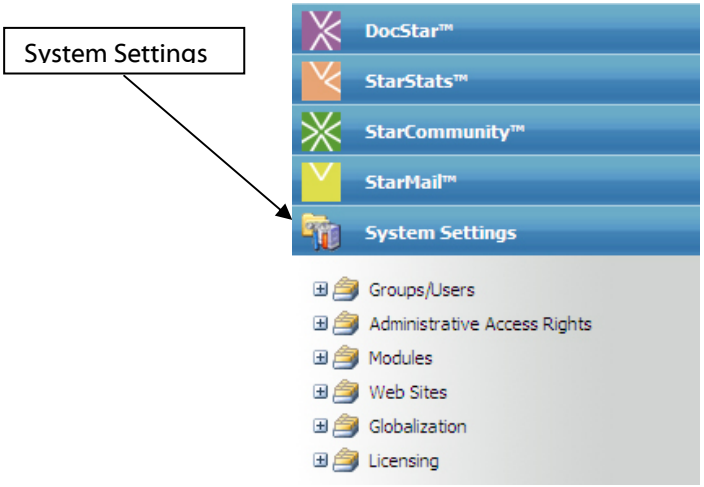
Below is an example of a StarMail configuration file.

```
<?xml version="1.0" standalone="yes"?>
<SettingsData xmlns="http://netstar.se/StarMail/SettingsData.xsd">
  <StarMail>
    <BrokerAddress>broker@starmail.netstar.se</BrokerAddress>
    <DefaultFromAddress>noreply@domain.com</DefaultFromAddress>
    <DefaultFromName>StarMail Sender Name</DefaultFromName>

<GoMailURL>http://www.domain.com/Netstar/StarMail/GoMail.aspx</GoMailURL>
    <StarMailFromAddress>name@domain.com</StarMailFromAddress>
    <MailServer>smtp.domain.se</MailServer>

<GoMailImageURL>http://www.domain.com/Netstar/StarMail/Images/pixel.gif</Go
MailImageURL>
    <Encoding>iso-8859-1</Encoding>
    <BrokerIP>195.178.179.55</BrokerIP>
```

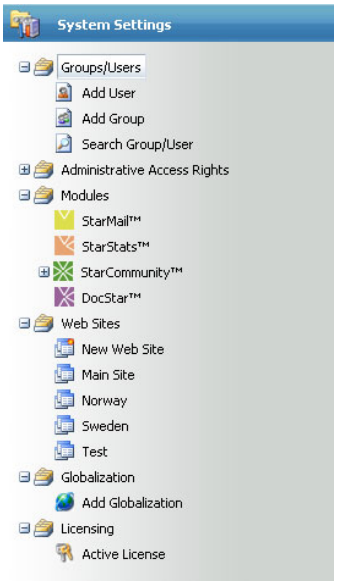
```
<BrokerPublicKey>
  <RSAKeyValue>
<Modulus>zG5Z4RaYhz5Grxlb+HVRWG2zzuGMECzngy1+OMmbOts0iZyirrZZlP69B3oh8xZJKA
Z/smE7WWx31uPasRmP2bWe9OTGbF13FwLv4OiEDuA5/M3Q6kn7Srdz/VhWURfB+CwoZkdgDv5t2
wY2tRRH4b33V1/qiJ7ZB20Cuuj566LdY+gIfJod3PSOIoE8VpQq1pPeof09U70oQHZKlf974BvB
jcVm4TCM5vR7G8hBZR0TAqf2rkNmby3BNAqkxpJnB9VSR6iec/XSLsisQZ39oeU2KAYgspLfi81
TyxG4ChyT/ar6Ao8MQHuapmkXG+IYANNMBAA20RP61UVrC1NfVTjjTRu5U5hMYqKB4btc3JPj9/
XgRtpMSZ+n961wuy3uXPFK86dWRlp0SNF6r++u/NB53wgmwrrWdsurYYsxCnE81gvK1CoPgKkFk
4hVBmuFytGem+Xuv3aDyA1ENtTgz7UV/DB2m3L64e6b1V4ukChzTUZXQmHejHfL21SXzbBW1/zY
AigpyhZke20LB66ndlMgb+jm8S5teftK1lpUnqKGd7/O17eQa47jMBdznZ2U1MDhnYcGxDJd+4J
OkW+smhRG3LrFDKfCOYIHtQRCw3OkXnL4TMjxub61V9U/AoknbamqXVlcgaUM3FKIR5Y3HayfGY
uUiOQI3PDEUsE4vSU=</Modulus>
    <Exponent>AQAB</Exponent>
  </RSAKeyValue>
</BrokerPublicKey>

  </StarMail>
</SettingsData>
```

5. System Settings

Here the StarMail administrator can create and remove groups or users that can help to administrate the StarMail.



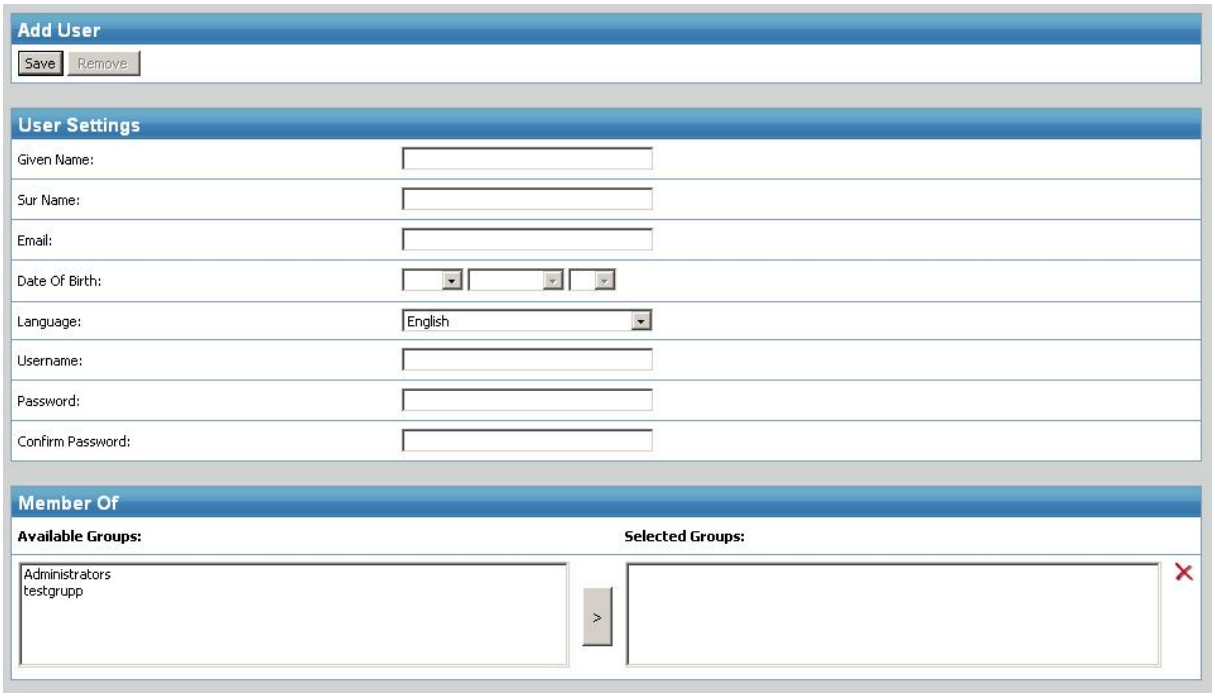
When System Settings is activated, the following menu is displayed



5.1. Users and User Groups

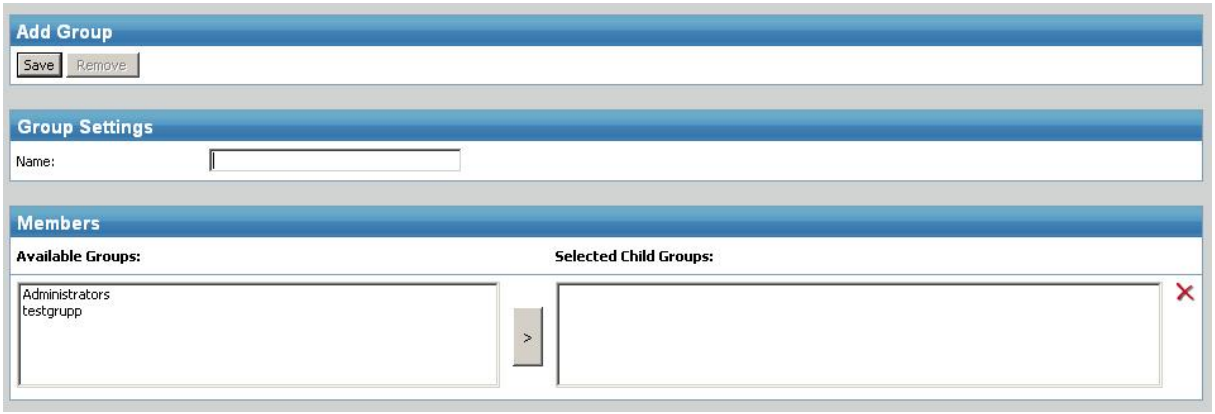
Add User

In the "Add User" page, you can create a new user with his basic details as well as add him to groups in order to inherit access rights.



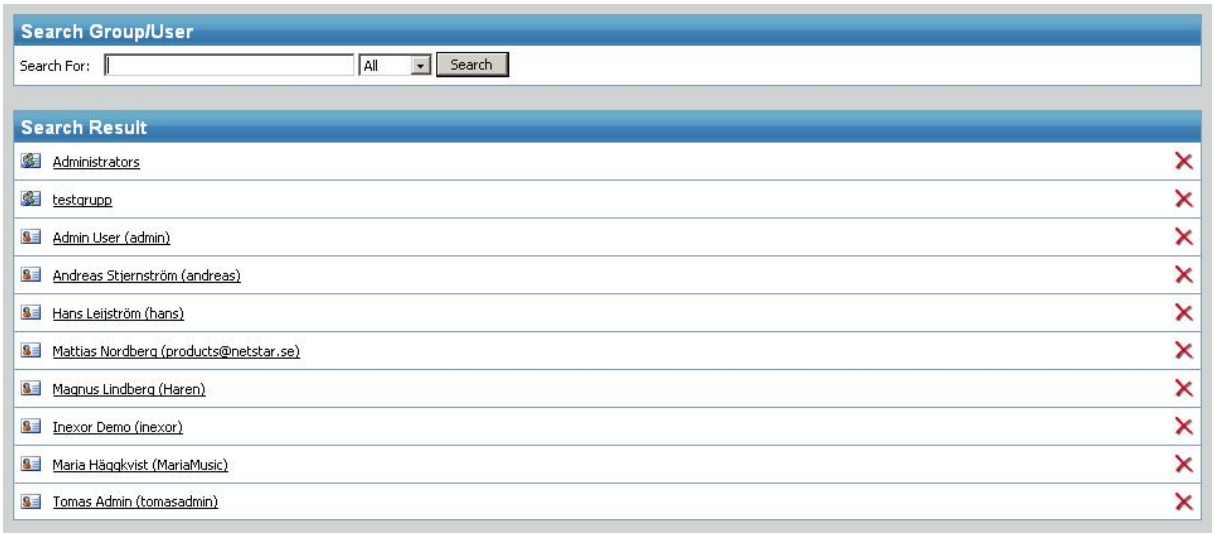
Add Group

A group it added by specifying its name. Groups can also have child groups that inherit access rights from their parent group.



Search

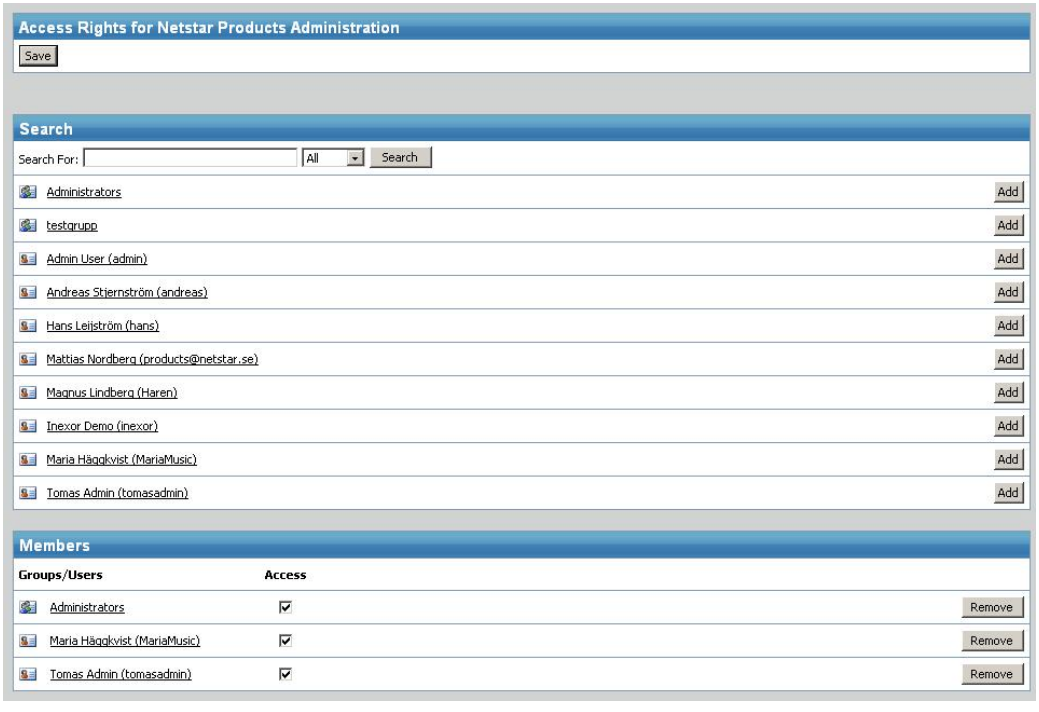
At the “Search Group/User” page you can search for both groups and users and remove them if needed.



5.2. Administrative rights

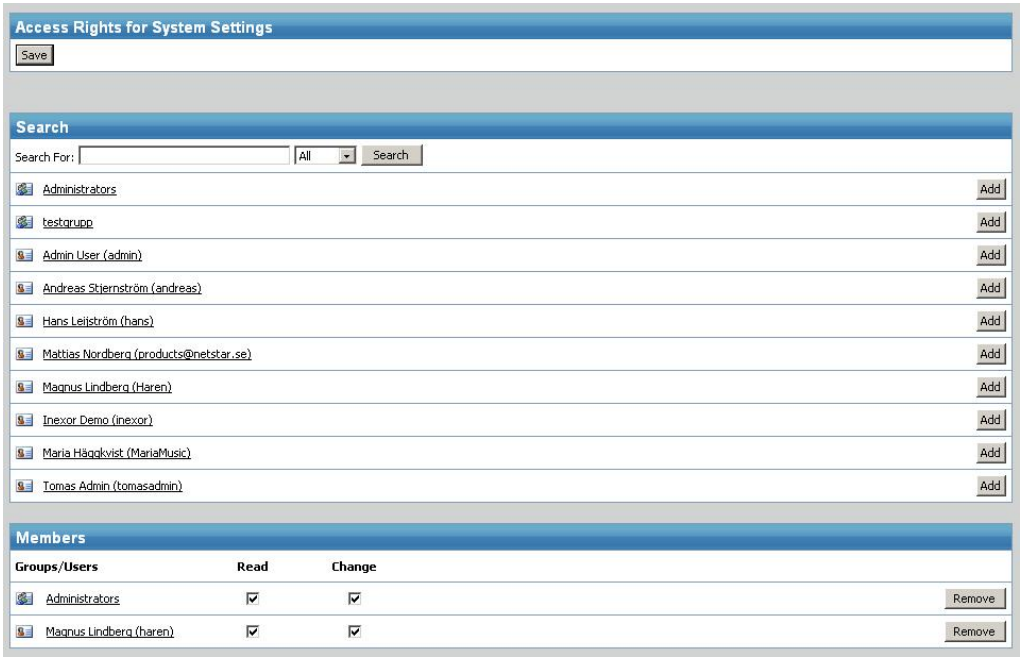
Administration

At the administration part of the access rights section the administrator can give users and groups rights to log in to Netstar Products Administration. To add a user or group, do a search for its name, click on "Add", check the checkbox "Access" followed by "Save". To remove a user or group, click "Remove" followed by "Save".



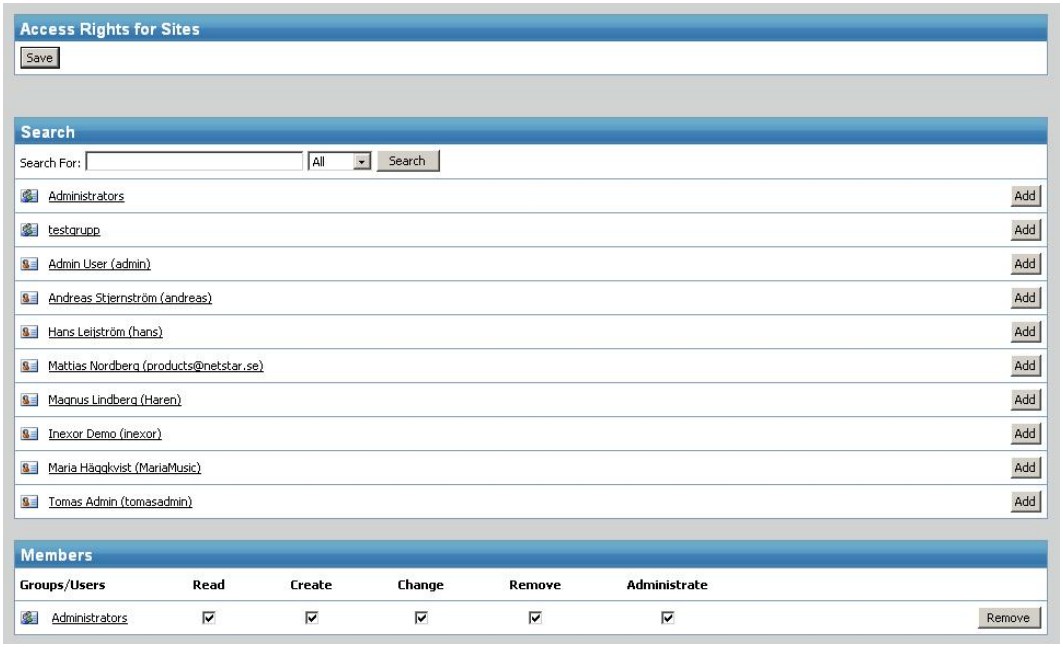
System Settings

The System Settings section contains the members and groups that have access to view the System Settings part of Netstar Product Administration. Users and groups can be added and removed in the same way as in the Administration section.



Web Sites

This section gives users and groups rights to view and do changes below the "Web Sites" section in System Settings. Adding and removing groups and users works in the same way as the other access right sections.

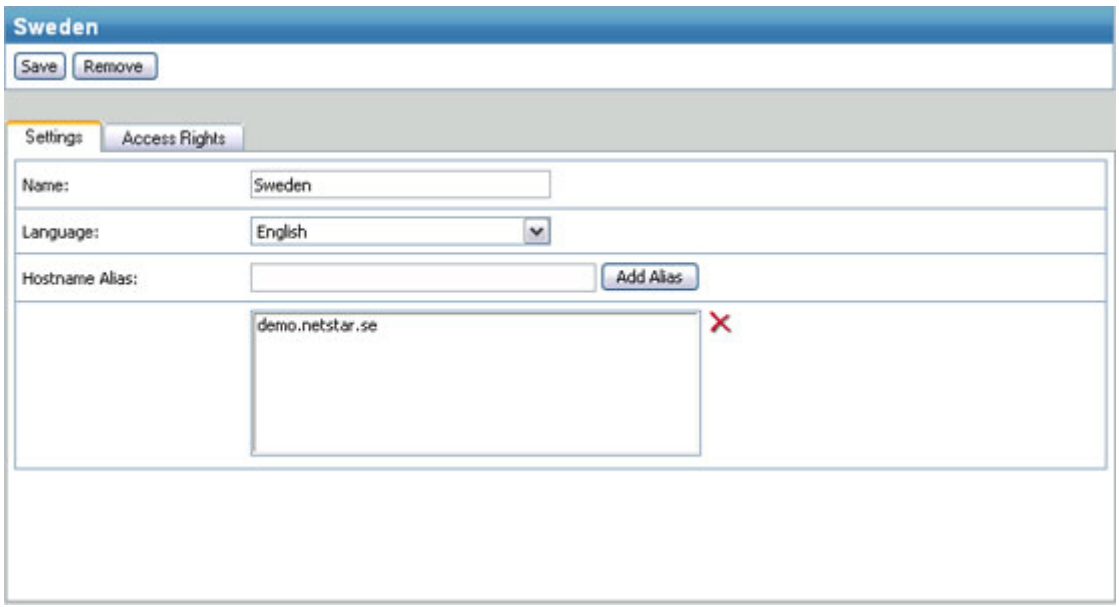


Security

At the security page, the administrator can give users and groups general security rights. This includes adding, changing and removing other users and groups.

5.3. Site Partitioning

Below the web sites header, new websites can be created as well as existing ones updated.

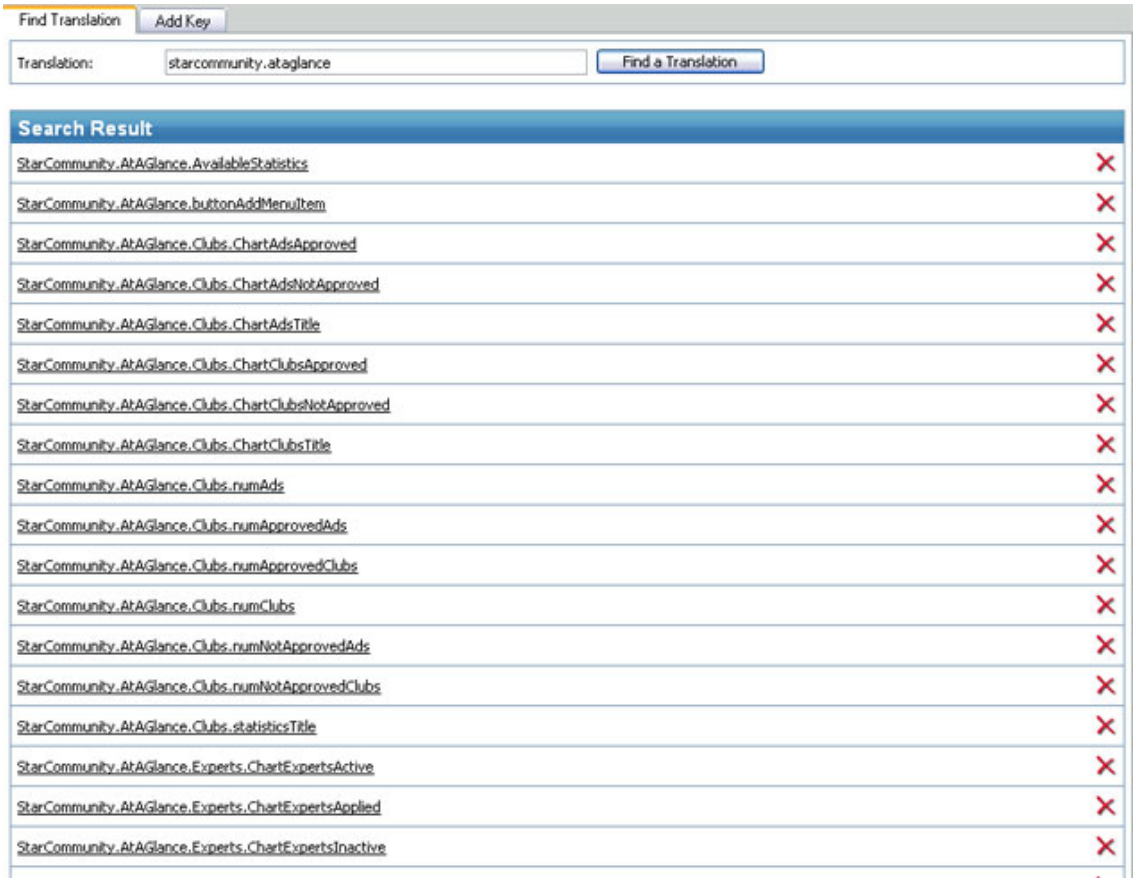


The administrator can choose new and the default language to use on the site. Hostnames are added by typing a name in the Hostname Alias textbox, followed by clicking on the “Add Alias” button.

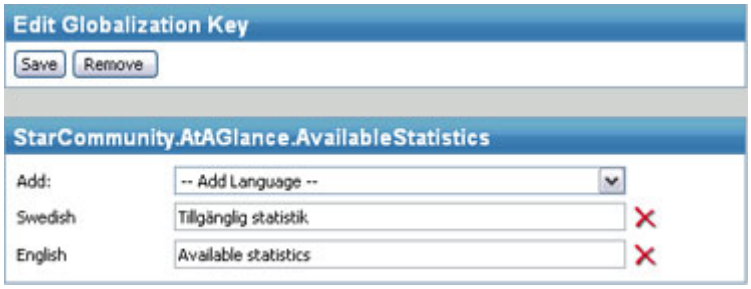
The Access Rights tab lets the administrator set what groups and users are allowed to operate on this site. The access rights available are Read, Create, Change, Remove and Administrate.

5.4. Globalization

The globalization section allows the administrator to translate texts in order to have web pages available in several different languages. At the "Find Translation" tab the administrator can search for language keys and edit or remove them. New keys can easily be added on the "Add Key" tab. To edit an existing key, search for the key and click on its name.

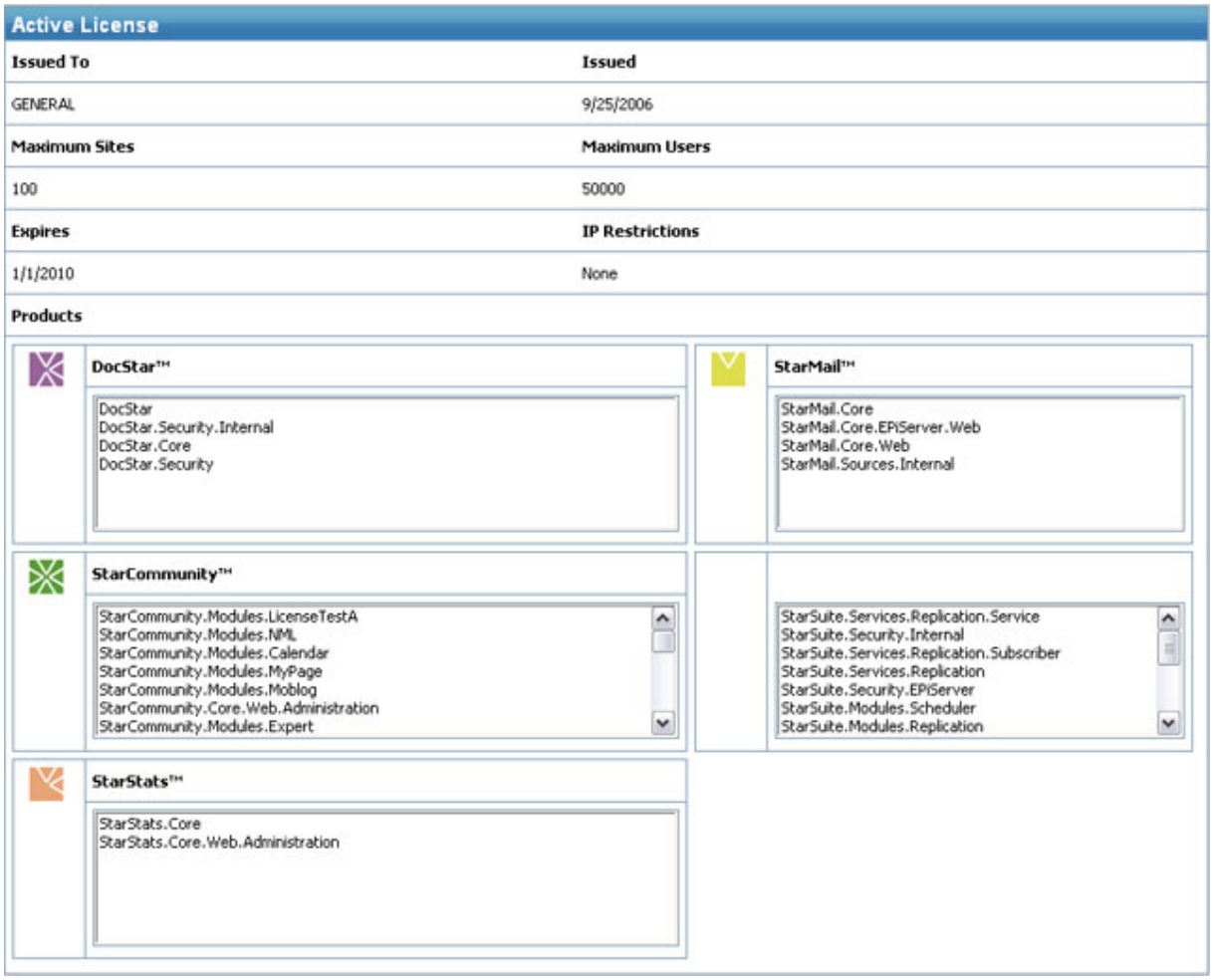


After having clicked on a key, the popup window is displayed, as shown on the screenshot below. This page displays the languages that the key has been translated to. Texts can be changed and languages can be added or removed.



5.5. Licensing

The “Active License” page contains information that is specific for your license. It lists all products, and their modules, that are installed on your system. The license has restrictions for maximum number of sites the administrator can create and maximum number of users that can register on the community. It also contains an expiration date for the whole license as well as IP restrictions on the servers that are hosting the system.



6. Requirements

6.1. Knowledge

StarMail requires basic knowledge of using:

- Internet Explorer / Firefox
- Development experience (to implement Recipient Sources)

6.2. System requirements

Client side (administrators)

- Internet Explorer 6 / Firefox 1.5

7. Related products

Other products from Netstar Products AB, from which web- and community systems may be based on, include:

- **StarCommunity** - software modules used to provide interactivity with target groups
- **DocStar** - web content management system used to manage all editorial web pages
- **StarStats** - for providing relevant event based statistics to support business decisions

8. Help us get better

Please send your comments and feedback on this user manual by email to product_feedback@netstar.se. We also appreciate if you notify us of any errors in this user manual or features you think are not described correctly.

9. Appendix 1. StarCommunity Source Sample Code

```
using System;
using System.Data;
using System.Data.Common;
using StarSuite.Core;
using StarSuite.Core.Data;
using StarSuite.Core.Modules;
using StarMail.Core;
using SC_Queries = StarCommunity.Core.Modules.Queries;
using SC_Security = StarCommunity.Core.Modules.Security;
using SC_Core = StarCommunity.Core;


namespace StarMail.Sources.StarCommunity
{
        /// <summary>
        ///  Basic recipient source for StarCommunity that has a recipient container per site
        /// </summary>
        public class StarCommunitySource : IRecipientSource
        {
                private Guid m_id;
                private string m_subscriptionUserAttributeName;

                public StarCommunitySource()
                {
                        m_id = new Guid("{07A3CA4C-EA15-444F-9569-C8B9916E31A2}");

                        LoadSettings();

            if (!string.IsNullOrEmpty(m_subscriptionUserAttributeName))
            {
                bool hasUserGuidAttribute = false;
                bool hasUserSubscribedAttribute = false;
                foreach (SC_Core.Modules.Attributes.IAttribute attribute in
SC_Core.Modules.Attributes.AttributeHandler.GetAttributes(typeof(SC_Core.Modules.Security.IUse
r)))
                {
                    if (attribute.Name == m_subscriptionUserAttributeName)
                        hasUserSubscribedAttribute = true;

                    if (attribute.Name == "StarMailUserGuid")
                        hasUserGuidAttribute = true;
                }

                if (!hasUserGuidAttribute)
                    SC_Core.Modules.Attributes.AttributeHandler.AddAttribute(new
SC_Core.Modules.Attributes.Attribute("StarMailUserGuid",
typeof(SC_Core.Modules.Security.IUser), typeof(string), true));

                if (!hasUserSubscribedAttribute)
                    SC_Core.Modules.Attributes.AttributeHandler.AddAttribute(new
SC_Core.Modules.Attributes.Attribute(m_subscriptionUserAttributeName,
typeof(SC_Core.Modules.Security.IUser), typeof(bool), false));
            }

                        StarSuite.Core.SiteHandler.SiteAdded += new
SiteEventHandler(SiteHandler_SiteAdded);
                        StarSuite.Core.SiteHandler.SiteRemoved += new
SiteEventHandler(SiteHandler_SiteRemoved);
                }

                #region IRecipientSource
                /// <summary>
                /// Gets the source's ID
                /// </summary>
                public Guid ID
                {
```

```
                        get
                        {
                                return m_id;
                        }
                }

                /// <summary>
                /// Gets whether the source supports importing lists
                /// </summary>
                public bool SupportsImport
                {
                        get
                        {
                                return false;
                        }
                }

                /// <summary>
                /// Gets the name of the source
                /// </summary>
                public string Name
                {
                        get
                        {
                                return "StarCommunity";
                        }
                }

                /// <summary>
                /// Import recipients, not supported for this source
                /// </summary>
                /// <param name="name"></param>
                /// <param name="importDelegate"></param>
                /// <param name="site"></param>
                /// <returns></returns>
                public RecipientContainer ImportRecipients(string name, RecipientImportDelegate
importDelegate, StarSuite.Core.Modules.ISite site)
                {
                        return null;
                }

                /// <summary>
                /// Gets all RecipientContainers for a specific site.
                /// </summary>
                /// <param name="site">The site</param>
                /// <returns>The RecipientContainers belonging to this site</returns>
                public RecipientContainerCollection
GetRecipientContainers(StarSuite.Core.Modules.ISite site)
                {
                        RecipientContainerCollection all = GetAllRecipientContainers();
                        RecipientContainerCollection rcc = new RecipientContainerCollection();
                        foreach (RecipientContainer rc in all)
                        {
                                if (rc.Site != null && site != null)
                                        if (rc.Site.ID == site.ID)
                                                rcc.Add(rc);

                                if (rc.Site == null && site == null)
                                        rcc.Add(rc);
                        }
                        return rcc;
                }

                /// <summary>
                /// Gets all global RecipientContainers
                /// </summary>
                /// <returns>The RecipientContainers</returns>
                public RecipientContainerCollection GetRecipientContainers()
                {
                        return GetRecipientContainers(null);
                }

                /// <summary>
```

```
                /// Gets a Recipient by its ID
                /// </summary>
                /// <param name="guid">The Recipient ID</param>
                /// <returns>The Recipient</returns>
                public Recipient GetRecipient(Guid guid)
                {
                        RecipientContainerCollection all = GetAllRecipientContainers();

                        // We have to check for this guid in each recipient container...
                        foreach (RecipientContainer rc in all)
                        {
                 SC_Queries.UserQuery uq = new SC_Queries.UserQuery();
                 SC_Queries.StringCriterion starMailUserGuid = new
SC_Queries.StringCriterion();
                 starMailUserGuid.Value = guid.ToString();
                 uq["StarMailUserGuid", rc.Site] = starMailUserGuid;
                 StarSuite.Core.Modules.Security.UserCollection uc =
SC_Core.StarCommunitySystem.CurrentContext.DefaultSecurity.GetQueryResult(uq);
                                if (uc.Count >0)
                                {
                                        SC_Security.IUser u = (SC_Security.IUser)uc[0];
                                        return GetRecipient(u, rc);
                                }
                        }
                        return null;
                }

                /// <summary>
                /// Gets the Recipients in specified RecipientContainer
                /// </summary>
                /// <param name="container">The RecipientContainer</param>
                /// <returns>The Recipients</returns>
                public RecipientCollection GetRecipients(RecipientContainer container)
                {
            bool isInTransaction = DatabaseHandler.InTransaction;
            if (!isInTransaction)
                DatabaseHandler.BeginTransaction();

            try
            {
                DatabaseHandler.ExecuteNonQuery(true,
"spStarMailUpdateStarCommunityUserGuids",
                            (container.Site != null ? (object)container.Site.ID : null),
m_subscriptionUserAttributeName);


                SC_Queries.UserQuery uq = new SC_Queries.UserQuery();
                SC_Queries.BooleanCriterion subscriptionCriterion = new
SC_Queries.BooleanCriterion();
                subscriptionCriterion.Value = true;
                uq[m_subscriptionUserAttributeName, container.Site] = subscriptionCriterion;

                StarSuite.Core.Modules.Security.UserCollection uc =
SC_Core.StarCommunitySystem.

CurrentContext.DefaultSecurity.GetQueryResult(uq);

                RecipientCollection rc = new RecipientCollection();
                foreach (SC_Security.IUser u in uc)
                {
                    Recipient r = GetRecipient(u, container);
                    if (r != null)
                        rc.Add(r);
                }
                return rc;
                if (!isInTransaction)
                    DatabaseHandler.Commit();
            }
            catch (Exception)
            {
                if (!isInTransaction)
                    DatabaseHandler.Rollback();
                throw;
```

```
            }
                }

                /// <summary>
                /// Sets unsubscribed status for specified Recipient
                /// </summary>
                /// <param name="recipient">The Recipient</param>
                /// <param name="isUnsubscribed">Whether the Recipient should be
unsubscribed</param>
                public void UnsubcribeRecipient(Recipient recipient, bool isUnsubscribed)
                {
                        RecipientContainer container =
GetRecipientContainer(recipient.ContainerId);

            SC_Queries.UserQuery uq = new SC_Queries.UserQuery();
            SC_Queries.StringCriterion starMailUserGuid = new SC_Queries.StringCriterion();
            starMailUserGuid.Value = recipient.ID.ToString();
            uq["StarMailUserGuid", container.Site] = starMailUserGuid;
            StarSuite.Core.Modules.Security.UserCollection uc =
SC_Core.StarCommunitySystem.CurrentContext.DefaultSecurity.GetQueryResult(uq);

            if (uc.Count >0)
                        {
                                SC_Security.IUser u = (SC_Security.IUser)uc[0];

                                if (isUnsubscribed)

        u.SetAttributeValue<bool>(m_subscriptionUserAttributeName, container.Site, false);
                                else
                        u.SetAttributeValue<bool>(m_subscriptionUserAttributeName, container.Site,
true);


        SC_Core.StarCommunitySystem.CurrentContext.DefaultSecurity.UpdateUser(u);
                        }
                }

                /// <summary>
                /// Gets a RecipientContainer by its ID
                /// </summary>
                /// <param name="guid">The RecipientContainer's ID</param>
                /// <returns>The RecipientContainer</returns>
                public RecipientContainer GetRecipientContainer(Guid guid)
                {
                        RecipientContainerCollection rcc = GetAllRecipientContainers();
                        foreach (RecipientContainer rc in rcc)
                        {
                                if (guid == rc.ID)
                                        return rc;
                        }
                        return null;
                }

                /// <summary>
                /// Gets a RecipientContainer by its name
                /// </summary>
                /// <param name="name">The RecipientContainer's name</param>
                /// <returns>The RecipientContainer</returns>
                public RecipientContainer GetRecipientContainer(string name)
                {
                        RecipientContainerCollection rcc = GetAllRecipientContainers();
                        foreach (RecipientContainer rc in rcc)
                        {
                                if (name == rc.Name)
                                        return rc;
                        }
                        return null;
                }

                /// <summary>
                /// Remove a RecipientContainer by its ID, this is not supported for this
source
                /// </summary>
```

```
                /// <param name="containerId">The RecipientContainer's ID</param>
                public void RemoveRecipientContainer(Guid containerId)
                {
                        return; // Can't remove this list.
                }
                #endregion IRecipientSource

                #region methods
                /// <summary>
                /// Gets all recipient containers
                /// </summary>
                /// <returns></returns>
                private RecipientContainerCollection GetAllRecipientContainers()
                {
            using (DbDataReader reader =
DatabaseHandler.GetReader("spStarMailGetStarCommunitySites"))
            {
                RecipientContainerCollection rcc = new RecipientContainerCollection();
                while (reader.Read())
                {
                    int siteId;
                    int recipientCount;
                    string name;

                    if (reader.IsDBNull(0))
                    {
                        siteId = -1;
                        recipientCount = GetRecipientCountBySite(null);
                        name = "All users";
                    }
                    else
                    {
                        siteId = reader.GetInt32(0);
                        Site site = StarSuite.Core.SiteHandler.GetSite(siteId);

                        if (site == null)
                            break;

                        if (((SC_Security.IUser)SC_Core.StarCommunitySystem.
                            CurrentContext.DefaultSecurity.NewUser).
                            GetAttribute(m_subscriptionUserAttributeName, site) == null)
                            break;

                        recipientCount = GetRecipientCountBySite(site);
                        name = "All users on " + site.Name;
                    }

                    rcc.Add(new RecipientContainer(reader.GetGuid(1), name, recipientCount,
                        m_id, DateTime.Now, siteId));
                }
                return rcc;
            }
                }

                /// <summary>
                /// Gets a Recipient by StarCommunity User and RecipientContainer
                /// </summary>
                /// <param name="user"></param>
                /// <param name="container"></param>
                /// <returns></returns>
                private Recipient GetRecipient(SC_Security.IUser user, RecipientContainer
container)
                {
                        AttributeCollection ac = new AttributeCollection();
                        ac.Add(new StarMail.Core.Attribute("First name", user.GivenName));
                        ac.Add(new StarMail.Core.Attribute("Last name", user.SurName));

                        bool unsubscribed =
!user.GetAttributeValue<bool>(m_subscriptionUserAttributeName, container.Site);

                        string userGuid = user.GetAttributeValue<string>("StarMailUserGuid",
container.Site);
                        if (userGuid != null)
```

```
                        {
                                Recipient r = new Recipient(new Guid(userGuid), user.EMail,
unsubscribed, container.ID, ac);
                                return r;
                        }
                        else
                                return null;
                }

                /// <summary>
                /// Gets Recipient count by Site
                /// </summary>
                /// <param name="site"></param>
                /// <returns></returns>
                private int GetRecipientCountBySite(Site site)
                {
            SC_Queries.UserQuery uq = new SC_Queries.UserQuery();
            SC_Queries.BooleanCriterion subscriptionCriterion = new
SC_Queries.BooleanCriterion();
            subscriptionCriterion.Value = true;
            uq[m_subscriptionUserAttributeName, site] = subscriptionCriterion;

            int count = 0;
            StarSuite.Core.Modules.Security.UserCollection uc = SC_Core.StarCommunitySystem.

CurrentContext.DefaultSecurity.GetQueryResult(uq, 1, 1, out count);

            return count;
                }

                /// <summary>
                /// Help method to load in the settings for this module.
                /// Will set the static variable upload file path.
                /// </summary>
                public void LoadSettings()
                {
                        SettingsData settings = new SettingsData();
                        string path =
System.IO.Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
                                @"Netstar\StarMail\StarMailStarCommunitySource.config");
                        settings.ReadXml(path);

                        if(settings.StarMailStarCommunitySource.Rows.Count > 0)
                        {
                                SettingsData.StarMailStarCommunitySourceRow starCommunitySource
= (SettingsData.StarMailStarCommunitySourceRow)settings.StarMailStarCommunitySource.Rows[0];
                                m_subscriptionUserAttributeName =
starCommunitySource.SubscriptionUserAttributeName;
                        }
                }

                /// <summary>
                /// Handles added sites
                /// </summary>
                /// <param name="args"></param>
                private void SiteHandler_SiteAdded(SiteEventArgs args)
                {
            bool isInTransaction = DatabaseHandler.InTransaction;
            if (!isInTransaction)
                DatabaseHandler.BeginTransaction();

            try
            {
                DatabaseHandler.ExecuteNonQuery("spStarMailAddStarCommunitySiteSource",
                                                args.Site.ID, m_subscriptionUserAttributeName);
                if (!isInTransaction)
                    DatabaseHandler.Commit();
            }
            catch (Exception)
            {
                if (!isInTransaction)
                    DatabaseHandler.Rollback();
                throw;
```

```
            }
        }

        /// <summary>
        /// Handles removed sites
        /// </summary>
        /// <param name="args"></param>
        private void SiteHandler_SiteRemoved(SiteEventArgs args)
        {
    bool isInTransaction = DatabaseHandler.InTransaction;
    if (!isInTransaction)
        DatabaseHandler.BeginTransaction();

    try
    {
        DatabaseHandler.ExecuteNonQuery("spStarMailRemoveStarCommunitySiteSource",
                                        args.Site.ID, m_subscriptionUserAttributeName);
        if (!isInTransaction)
            DatabaseHandler.Commit();
    }
    catch (Exception)
    {
        if (!isInTransaction)
            DatabaseHandler.Rollback();
        throw;
    }
        }
        #endregion methods
    }
}
```